Exhibit 1

IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

HEADWATER RESEARCH LLC

Plaintiff,

v.

SAMSUNG ELECTRONICS CO., LTD., SAMSUNG ELECTRONICS AMERICA, INC.

Defendants.

Case No. 2:23-CV-00103-JRG-RSP

REBUTTAL EXPERT REPORT OF ERIK DE LA IGLESIA REGARDING VALIDITY

Date: October 14, 2024

Erik de la Iglesia

TABLE OF CONTENTS

B. Lee Alone or in Combination Does Not Render Obvious the Asserted Claims

239. Dr. Foster references IPR2024-00003, as supposedly showing the obviousness of the '117 Patent. Dr. Foster says he "agree[s] with the analysis of the IPR petition and its supporting materials," and says he "reproduce[s] some of that analysis below and adopt[s] it as [his] own opinions." Foster Rpt. ¶ 552. However, what Dr. Foster neglects to acknowledge is that the Patent Trial and Appeal Board of the United States Patent and Trademark Office already considered and disagreed with the analysis in IPR2024-00003. *See* IPR2024-00003 at Paper No. 8.

240. Furthermore, there are numerous reasons why Lee alone or in combination with other references does not disclose or render obvious the asserted claims, as explained in further detail below.

241. Moreover, there is significant evidence of secondary considerations (e.g., commercial success, long-felt need, failures of others, praise, awards, industry recognition, copying) that further support my opinions that the asserted claims would not have been obvious to a POSITA, which are discussed below and hereby incorporated by reference.

1. ***Claim 1[d]: "the network message server to generate corresponding Internet data messages based on the requests, each such message containing at least one application identifier for an indicated application and application data corresponding to one of the requests."***

242. Claim limitation 1[d] recites: "the network message server to generate corresponding Internet data messages based on the requests, each such message containing at least one application identifier for an indicated application and application data corresponding to one of the requests."

243. Dr. Foster acknowledges that a requirement of this limitation is that "'each' message generated by the network message server [must] contain an 'application identifier' and 'application data.'" Foster Rpt. ¶ 585. I disagree, however, that Lee teaches or renders obvious this requirement.

244. For example, Dr. Foster does not provide any express teaching from Lee that "each" message would include application data. Thus, there is no dispute that Lee does ***not*** actually teach

as much.

245. His only allegation that this requirement would have been obvious is that "a message with only an application identifier and no application data would be pointless." Foster Rpt. ¶ 585. However, that assertion is, in my opinion, based on a misunderstanding of how the Lee system worked to "invoke" applications.

246. Specifically, Lee teaches "selectively invoke[ing] the application 260 corresponding to the associated application ID"—in other words, it tells the application to begin executing so that the application can itself retrieve the message data. Lee ¶[29]. "Invoking," in the context of computer software processes, an application does not mean "forwarding data to the application," but instead means ***causing the application to execute***. By ***selectively*** invoking applications in this manner, it "reduc[es] a number of the applications resident in memory," thereby reducing "power consumption." *Id.*; *see also id.* FIG. 8 (showing the integrated push service agent "invok[ing] application corresponding to app_ID").

247. Thus, Lee uses the application ID to inform applications that they should begin running. A POSITA would recognize that an application that receives an invocation request with no application ID pursuant to a push message could be programmed to contact its associated application server through a direct connection to retrieve any needed data from that server without the push client needing to act as an intermediary.

248. For example, an important use-case for such an implementation would be if the application server wanted to provide highly sensitive information to an application, and wished to use a ***direct connection*** to provide that information. Normally, this is difficult to accomplish, because the application might not be running on the device. But Lee's application invocation process allows a way for the application server to request that the relevant application be ***invoked***, and the application under such a scenario would be programmed to contact the application server. No application data would need to be transmitted along with the invocation message, because contacting the application server upon invocation would be part of the application's programming.

249. One example of this implementation (which is consistent with Lee's teachings) is

a scenario where a bank application server needs to transmit data securely to a user's banking application, but wants to do so over a direct connection. For instance, if the Chase banking network had an important security alert for the user, it could simply inform Lee's system to invoke the Chase banking application (without actually providing application data to the application); the banking app would then contact the Chase banking network server upon invocation, and the network server could securely inform the banking app of the security alert. This would avoid the need for complex security mechanisms, and would not involve the invocation message sent to the application including any application data.

2. ***Claim 1[g]: [each device messaging agent, when executing,] to, for each received message, map the application identifier in the message to a software process corresponding to the application identifier, and forward the application data in the message to the software process via a secure interprocess communication service.***

a. No motivation to combine with Kalibjian.

250. Dr. Foster's proposed combination of Kalibjian, memory in the device is "divided into 'separate and distinct secure sections.'" Foster Rpt. ¶ 596. Under the proposed combination, each application would need to operate in a separate partition from other applications, such that no application could potentially have access to information intended for another application (or corrupt other applications). *See* Kalibjian ¶[0020] ("A secure operating system includes secure memory partitions or processing space that insures applications running in one partition 10A cannot gain access and corrupt applications running in another partition 210B."); Foster Rpt. ¶ 617 ("In the Lee-Kalibjian combination, the push client would have been incorporated into one of these secure sections, …. with applications incorporated into other secure sections….").

251. Dr. Foster provides two alleged motivations to do so. His first motivation is "to address 'security concerns' such as different applications running on the mobile terminal intercepting or interfering with messages exchanged between Lee's push service agent 250… and application 230." Foster Rpt. ¶ 597.

252. This motivation (avoiding intercepting or interfering with messages) fails because he does not allege that Lee's mobile device suffered from any such problem. (I note that neither